Applicant : von Bergen et al.
Serial No. : 10/625,908
Filed : July 24, 2003
Page : 6 of 9

Attorney's Docket No.: 13909-119001

REMARKS

This application has been carefully reviewed in light of the Office Action dated June 2, 2006. Claims 1 to 24 are in the application, of which claims 1, 11, and 21, the independent claims, have been amended herein. Reconsideration and further examination are respectfully requested.

Initially, the Examiner's indication that the objection to claim 1, the double patenting rejection, and the rejection under 35 U.S.C. § 101 have been withdrawn is acknowledged with appreciation.

In the Office Action, claims 1 to 24 were rejected under 35 U.S.C. § 102(b) over U.S. Patent No. 5,784,699 ("McMahon"). In response, the independent claims have been amended to further clarify the features that *i)* the block of memory is divided into a plurality of frames, with each of the plurality of frames operable to store an indexing structure associated with an attribute of a data record, and *ii)* each of the frames is divided into a plurality of instances, with each of the plurality of instances operable to store an index node of the indexing structure. Support for these newly clarified features is found throughout the specification, including at least the paragraphs that begin on line 8 of page 7, and line 10 of page 11. Withdrawal of the rejection and further examination are respectfully requested.

The present disclosure generally relates to memory allocation. A block of memory is divided into a plurality of frames, with each of the plurality of frames operable to store an indexing structure associated with an attribute of a data record, and each of the frames is divided into a plurality of instances, with each of the plurality of instances operable to store an index node of the indexing structure. Furthermore, data elements are stored, or populated, in the plurality of instances.

Referring to particular claim language, independent claim 1 recites a method for allocating memory in a computer system. The method includes outputting a request from an application to an operating system for allocation of a block of memory by the operating system to the application, and accessing the block of memory for the application. The method also includes dividing the block of memory into a plurality of frames, with each of the plurality of

frames operable to store an indexing structure associated with an attribute of a data record, and dividing each of the frames into a plurality of instances, with each of the plurality of instances operable to store an index node of the indexing structure. The method further includes associating the attribute with the plurality of instances for data storage using the plurality of instances.

Independent claim 11 recites a software application tangibly embodied on a computer-readable medium using application-level memory management. The software application includes an application-level memory manager operable to allocate a block of memory to store data elements, and to divide the block of memory into a plurality of frames, with each of the plurality of frames operable to store an indexing structure associated with an attribute of a data record, and further to divide each frame into a plurality of instances, with each of the plurality of instances operable to store an index node of the indexing structure. The software application also includes application code operable to associate the attribute with the plurality of instances for storage of the data elements in the plurality of instances.

Independent claim 21 recites a method including associating data elements used by an application with an application-defined instance type. The method also includes associating the application-determined instance type with an application-determined one of a plurality of blocks of memory allocated by an operating system, wherein the application-determined memory block is divided into a plurality of frames, with each of the plurality of frames operable to store an indexing structure associated with an attribute of a data record, the plurality of frames being further divided into a plurality of instances, with each of the plurality of instances operable to store an index node of the indexing structure. The method further includes populating the plurality of instances with the data elements.

The applied art is not seen to disclose, teach, or to suggest the foregoing features recited by the independent claims. In particular, McMahon is not seen to disclose at least the features that *i)* the block of memory is divided into a plurality of frames, with each of the plurality of frames operable to store an indexing structure associated with an attribute of a data record, and *ii)* each of the frames is divided into a plurality of instances, with each of the plurality of instances operable to store an index node of the indexing structure.

Applicant : von Bergen et al.
Serial No. : 10/625,908
Filed   : July 24, 2003
Page   : 8 of 9

Attorney's Docket No.: 13909-119001

McMahon describes a computer system which implements a memory allocator that employs a data structure to maintain an inventory of dynamically allocated memory available to receive new data. *See* McMahon, col. 4, ln. 58 to col. 5, ln. 8; and Abstract. Although the Office Action, at page 10, alleges that the free list discloses the feature of identifying unused *instances*, where *instances* are <u>within</u> a block of memory, the applicants again respectfully disagree. Clearly, the passage beginning at column 5, line 25 of McMahon is seen to describe using a free list to search for an available memory *block*, and not an available *instance* <u>within</u> a block. In particular, the passage states (emphasis added):

> In operation, the dynamic memory allocator 50 receives requests from a software program. As shown in block 110, the dynamic memory allocator 50 rounds the memory request size to the nearest bin size in accordance with the predetermined rounding factor. The dynamic memory allocator 50 *searches a free list*, corresponding to the bin size, *for an available memory block* as shown in block 120. If a *memory block is available* for the appropriate bin size, then the dynamic memory allocator 50 *assigns a memory block* utilizing the pointer on the free list as shown in blocks 130 and 135. As shown in blocks 130 and 140 on FIG. 2, *if a memory block is not available* in the specified bin, then the dynamic memory allocator 50 searches for an *available memory block* from a free list that corresponds to the next largest bin size. (McMahon, col. 5, ll. 25 to 39).

Accordingly, the cited passage is not seen to describe searching for anything other than an available memory *block*, and is certainly not seen to describe the feature of identifying available, or unused *instances* <u>within</u> a memory block. Furthermore, McMahon is not seen to describe, nor does the Office Action even assert that McMahon describes, the newly clarified features of the independent claims, that *i)* the block of memory is divided into a plurality of frames, with each of the plurality of frames operable to store an indexing structure associated with an attribute of a data record, and *ii)* each of the frames is divided into a plurality of instances, with each of the plurality of instances operable to store an index node of the indexing structure.

Accordingly, based on the foregoing amendments and remarks, independent claims 1, 11 and 21 are believed to be allowable over the applied references. The remaining rejected claims in the application are each dependent on these independent claims and are believed to be allowable for at least the same reasons. Because each dependent claim is deemed to define an additional aspect of the invention, individual consideration of each on its own merits is respectfully requested.

Applicant : von Bergen et al.
Serial No. : 10/625,908
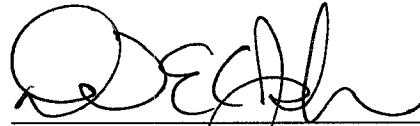Filed : July 24, 2003
Page : 9 of 9

Attorney's Docket No.: 13909-119001

No other matters being raised, it is believed that the entire application is fully in condition for allowance and such action is courteously solicited.

No fees are believed to be due at this time. Please apply any other charges or credits to deposit account 06 1050.

Respectfully submitted,

Date: June 28, 2006

David E. A. Jordan
Reg. No. 50,325

Fish & Richardson P.C.
1425 K Street, N.W.
11th Floor
Washington, DC 20005-3500
Telephone: (202) 783-5070
Facsimile: (202) 783-2331

40351882.doc